AL-TP-1992-0035-Vol-1

AD-A256 752

(2)

FUNCTIONAL REQUIREMENTS OF AN ADVANCED INSTRUCTIONAL DESIGN ADVISOR: EPITOMIZING FUNCTIONS (VOLUME 1 OF 3)

ARMSTRONG LABORATORY

Robert D. Tennyson

Department of Educational Psychology
University of Minnesota
Minneapolis, MN 55455

Charles R. Reigeluth

Department of Educational Psychology
Indiana University
Bloomington, IN 47405

Dennis J. Gettman

HUMAN RESOURCES DIRECTORATE
TECHNICAL TRAINING RESEARCH DIVISION
Brooks Air Force Base, TX 78235-5000

September 1992

Interim Technical Paper for Period February 1991 – May 1992

Approved for public release; distribution is unlimited.

AIR FORCE MATERIEL COMMAND
BROOKS AIR FORCE BASE, TEXAS 78235-5000

NOTICES

This technical paper is published as received and has not been edited by the technical editing staff of the Armstrong Laboratory.

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the Government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

The Office of Public Affairs has reviewed this paper, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

This paper has been reviewed and is approved for publication.

DENNIS J. GETTMAN
Project Scientist

HENDRICK W. RUCK, Technical Director
Technical Training Research Division

RODGER D. BALLENTINE, Colonel, USAF
Chief, Technical Training Research Division

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
September 1992

3. REPORT TYPE AND DATES COVERED
Interim – February 1991 – May 1992

4. TITLE AND SUBTITLE
Functional Requirements of an Advanced Instructional Design Advisor: Epitomizing Functions (Volume 1 of 3)

5. FUNDING NUMBERS
C – F33615-88-C-0003
PE – 62205F
PR – 1121
TA – 10
WU – 43

6. AUTHOR(S)
Robert D. Tennyson
Charles R. Reigeluth
Dennis J. Gettman

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Department of Educational Psychology
University of Minnesota
Minneapolis, MN 55455

Department of Educational Psychology
Indiana University
Bloomington, IN 47405

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)
Armstrong Laboratory
Human Resources Directorate
Technical Training Research Division
Brooks Air Force Base, TX 78235-5000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AL-TP-1992-0035-Vol-1

11. SUPPLEMENTARY NOTES
Armstrong Laboratory Technical Monitor: Dr. Daniel J. Muraida, (512) 536-2981

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The Advanced Instructional Design Advisor (AIDA) will provide automated and intelligent assistance to inexperienced computer-based instruction developers. This report outlines initial development of a prototype or XAIDA as opposed to development of a complete automated Instructional System Design (ISD) expert system. One of the difficulties in automating instructional design is with the proper application of ISD. The first part of this paper offers a detailed description of a potential ISD model for XAIDA in which three basic forms of knowledge, declarative, procedural, and contextual, are directly tied to instructional strategies. The final section of the paper specifies conditions for the development of XAIDA. This "epitomizes" XAIDA by presenting general "simplifying conditions" to restrict such things as task, time to teach, learning level, medium, environment, and instructional strategy. The final portion of this paper applies these general conditions to a specific task by outlining the development of a lesson for the T-38 engine starting system.

14. SUBJECT TERMS
Advanced Instructional Design Advisor
Computer-based instruction
Computer-based training
ISD

15. NUMBER OF PAGES
66

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

TABLE OF CONTENTS

LIST OF FIGURES

FREFACE

The work reported herein was done for the Advanced Instructional Design Advisor project at the Air Force Armstrong Laboratory (Human Resources Directorate). The substance of this research was done under contract to Mei Associates, Inc., the primary contractor on the Advanced Instructional Design Advisor (Contract No. F33615-88-C-0003).

This work was done as part of the second phase effort on the Advanced Instructional Design Advisor. The initial phase of this project established the conceptual framework and functional specifications for the Advanced Instructional Design Advisor, an automated and intelligent collection of tools to assist subject matter experts who have no special training in instructional technology in the design and development of effective computer-based instructional materials. This second phase provided the design specifications for an experimental prototype.

Mei Associates' final report for the second phase was published as Armstrong Laboratory Technical Paper AL-TR-1991-0085. In addition, Mei Associates received nine papers from various consultants working on this phase of the project. These nine papers have been grouped into 3 sets and edited by AL/HRTC personnel. They are published as Volumes 1 - 3 of Functional Requirements of an Advanced Instructional Design Advisor:

Volume 1: Epitomizing Functions

Volume 2: Task Analysis and Troubleshooting

Volume 3: Simulation Authoring

This is Volume 1 in the series. Mr Dennis Gettman wrote Sections I and IV. Dr Robert D. Tennyson wrote Section II and Dr Charles R. Reigeluth wrote section III.

SUMMARY

The Advanced Instructional Design Advisor is an R & D project being conducted by the Air Force Human Resources Laboratory in response to an Air Training Command (ATC) Manpower, Personnel, and Training Need calling for improved guidelines for authoring computer-based instruction (CBI) (MPTN 89-14T).

Aggravating the expensive and time-consuming process of CBI development is the lack of Air Force personnel who are well-trained in the areas of instructional technology and educational psychology. More often than not, a subject matter expert with little knowledge of CBI is given the task of designing and developing a computer-based course. Instructional strategies that work in a classroom are often inappropriate in a computer-based setting (e.g., leading questions may work well in a classroom but are difficult to handle in a computer setting). Likewise, the computer offers the capability to present instruction in ways that are not possible in the classroom (e.g., computer simulations can be used to enhance CBI).

The Advanced Instructional Design Advisor is a project aimed at providing subject matter experts who have no background in computer-based instructional systems with automated and intelligent assistance in the design and development of CBI. The goal is to reduce CBI development time while insuring that the instructional materials are effective.

I. INTRODUCTION (Gettman)

The Advanced Instructional Design Advisor is an R & D project aimed at providing automated and intelligent assistance to inexperienced instructional designers who have the task of designing and developing computer-based instruction (CBI). The particular problem being addressed by this line of research is the need for more cost efficient methodologies for the design and development of CBI. Current methods for developing CBI are expensive, time-consuming, and often result in ineffective instruction due to the general lack of expertise in computer-based instructional systems (Spector, 1990).

The Advanced Instructional Design Advisor project is divided into four phases:

Phase 1: Conceptualization & Functional Specifications

Phase 2: Conceptual Refinement & System Specifications

Phase 3: Prototype, Field Test, & Refinement

Phase 4: Technology Demonstration & System Validation

The first two phases have been performed under Task Order Contracts. The third phase is being accomplished via a Broad Agency Announcement (BAA). The fourth phase will be funded by a fully specified contract. The work reported herein concerns the second phase.

The next two sections of this report outline initial development of a prototype AIDA. In Section II Tennyson provides rationale for the development of a prototype or XAIDA as opposed to development of a complete automated ISD expert system. Also in Section II Tennyson gives a detailed description of his recommended ISD model for XAIDA in which three basic forms of knowledge; declarative, procedural, and contextual, are directly tied to instructional strategies.

In Section III Reigeluth specifies conditions, in general (Part I) and specifically (Part II), for the development of an XAIDA prototype. Reigeluth "epitomizes" the XAIDA by presenting functions which the prototype should include by describing general "simplifying conditions" to restrict such things as: task, time to teach, learning level, medium, environment, and instructional strategy. Reigeluth applies these general conditions to a specific task in Section IV by outlining the development of a lesson for the T-38 engine starting system.

II. AN AIDA PROTOTYPE (Tennyson)

Background

This report presents a modified ISD model as the basic instructional development effort for a prototype AIDA. Proposed is model containing three basic sets of authoring activities: identification of learning objectives, analysis of the information to be learned, and instructional strategies. The modified ISD model is constructed as the first minimal set of authoring activities taken from an incremental approach to the design of AIDA. Subsequent versions of AIDA would reflect additional layers of ISD authoring activities. The proposed modified model represents a clear departure from the current ISD models by employing both advancements from cognitive science and computer/interactive technologies. Along with the presentation of the model, an example is included to illustrate the application of the model within the AIDA environment.

In my Phase 1, Cycle 2 technical report (Tennyson, 1990b), I presented framework specifications for a complete expert system for automating instructional development in the AIDA environment. To produce such a system, two possible approaches were discussed. The first approach would be to develop the ISD expert system for AIDA as presented. The second would be to follow an incremental approach in which an initial prototype is developed that only has a minimal set of ISD features and is aimed at an experienced courseware author. That is, an ISD expert system that would only have an advisor level tutor and the situational evaluation and

recommendations components (see Tennyson, 1990b, Figure 2). The content knowledge base and acquisition features of the intelligent interface tutor and the production component would be added in subsequent iterations.

Although the first approach seems possible, I raised three important problems that should be considered and that would possibly favor the second approach. The first problem involves the cost and time necessary to produce an expert system. For example, given the need to produce expert systems that are both timely and profitable, the majority of expert systems are developed using commercially-available skeleton shells. Although AIDA does not directly need to generate income within a reasonable time-frame, it does have the constraints of a budget that limits writing the system from scratch. Because of this problem, I recommended rapid prototyping of AIDA by employing commercial shells that are linked by some general language (Hewett, 1989). Thereby, instead of five years of time and a multi-million dollar budget to produce a complete version of AIDA, an initial prototype (AIDA1) could be developed in much less time and within the constraints of a limited budget.

A second major problem I raised has to do with the learning theory paradigm shift (from behavioral to cognitive) for the instructional theory necessary to support the rule structure of AIDA's expert system. There has been minimal empirical research to date on instructional variables and conditions associated with the extension of cognitive learning theory to instructional

design theory. Even though it is possible to develop an initial AIDA prototype with a limited set of authoring activities, additional research in cognitive-based instructional design theory (including research on the effects of computer/interactive technologies) needs to be done for any future elaborations of AIDA. A third problem area I identified relates to the specification of the human-computer interaction principles necessary to run and manage the complex environment of an automated ISD expert system.

Within the constraints defined above for approach one, Tennyson recommended the following incremental approach to the development of AIDA:

An Incremental Approach for AIDA Development

1. Framework specifications. This step conceptualizes the idea or vision of the expert system. My Cycle 2 report serves as an example of this first activity in producing an automated instructional development system.

2. Functional specifications. From the initial outline of the basic system, the specific functions provided by the system need to be defined. From this step a rapid prototype can be developed as follows:

-Write functional specifications;

-Summarize what is known/not known about the functions;

-Estimate the complexity of the functions;

-Based on the summary and estimates, group the functions into AIDA1 (i.e., Phase II, Cycle 1 AIDA prototype), and then

prioritize the functions for successive implementations for versions AIDA2, AIDA3, etc. Each version would add layers of functions and increased use of a high-level computer language.

3. Logical design. Starting with AIDA1, define the logical components that provide the specified functions.

4. Physical design. Define the software modules which implement the logical design of the system.

5. Programming. With AIDA1, rapid software development is recommended while with the successive versions, the software procedures defined by my Cycle 2 technical report should be followed.

6. Testing. Once AIDA1 is developed, it should be tested following standard computer software benchmark criteria (e.g., O'Neil's report).

7. Implementation. Complete the remaining tasks to implement AIDA1 while simultaneously accumulating the experience and research findings needed to produce AIDA2, etc.

8. Incremental development. Basically starting with number 3 above, iteratively build AIDA towards a system that includes all of the functions defined in numbers one and two.

A Modified ISD Model

In my Phase I, Cycle 1 technical report, I present an updated ISD model based upon advancements from cognitive science and educational technology. This updated model, labelled the fourth generation, included a complete reference to all aspects of instructional systems development, including not only

identification of authoring activities associated with courseware development but other non-instructional activities (e.g., test development, media production). As discussed above, an incremental approach to the development of AIDA would begin with a basic set of authoring activities. In this report, I propose a set of authoring activities that would make-up AIDA1.

This basic set follows from the 4th generation ISD model and the Tennyson and Rasch model (1988; 1990). The Tennyson and Rasch model link desired learning objectives with specific instructional strategies. Validity of this model is in part supported by Gagne and Merrill's (1990) thesis on integrated objectives. Relying on these two primary sources (my two Phase 1 reports), I am proposing that a modified ISD model operate at an advisory level (i.e., to be used by experienced courseware developers) and that it include both the situational evaluation and recommendations components. For the authoring activities, AIDA1 would have a minimum set of activities associated with identifying desired objectives, an analysis of the information to be learned, and instructional variables to support integrated instructional strategies.

In the following subsections I will present a basic set of authoring activities to support AIDA1. To help understand the activities, I will also present an example illustrating the application in an adult learning situation. The three subsections are learning objectives, information analysis, and instructional strategy prescriptions. These three areas would

constitute an initial set of ISD authoring activities for AIDA1, the prototype. Before presenting the three sets of authoring activities, I will briefly discuss the relationship of cognitive learning to objectives and instructional strategies. I will do this using a modified version of the Tennyson and Rasch (1988) model.

Figure 1, shows the direct integration of cognitive learning theory with learning objectives and prescribed instructional strategies.

ISD Components	Acquisition of Knowledge Base		
Memory Systems	Declarative Knowledge	Procedural Knowledge	Contextual Knowledge
Learning Objectives	Verbal Information	Intellectual Skills	Contextual Skills
Instructional Strategies	Expository Strategies	Practice Strategies	Problem-Oriented Strategies

Figure 1. Instructional design model linking cognitive learning theory with learning objectives and instructional strategies.

Memory Systems

The proposed modified model is directly associated to a cognitive paradigm of learning. This paradigm is discussion in my article, "A Proposed Cognitive Paradigm of Learning for Educational Technology" (Tennyson, 1990a). Because the purpose of this report is with improvement in the acquisition of knowledge, only the storage system of long-term memory is discussed. The storage system is composed of three basic forms of knowledge: Declarative knowledge, knowing "that" about the information; procedural knowledge, knowing "how" to use information; and, contextual knowledge, knowing "when and why" to use given information.

Proposed in the modified model (see Figure 1) is that there is a direct connection between the three basic types of knowledge and prescribed instructional strategies. The purpose for including this component in the model is twofold: First, to establish a direct link between instructional theory and learning theory: This was done successfully with the behavioral paradigm where instructional strategies were designed following the conditions of that paradigm. Thus, I have attempted in this report to make an association between the cognitive paradigm and instructional strategies. And, second, to indicate the relative strengths of the instructional strategies in reference to the types of knowledge. Within the proposed modified model, the learning objectives tie directly with the memory systems component with specific instructional strategy prescriptions.

Learning Objectives

The purpose of cognitive-based learning objectives is to further elaborate the curricular goal of knowledge acquisition. Objectives are important in the planning of learning environments because they provide the means for identifying specific instructional strategies. I define three basic learning objectives for the knowledge acquisition domain as follows:

- Verbal information. This objective deals with the learner acquiring an awareness and understanding of the concepts, rules, and principles within a specified domain of information (i.e., declarative knowledge).
- Intellectual skills. This objective involves the learner acquiring the skill to correctly use the concepts, rules, and principles of a specified domain of information (i.e., procedural knowledge).
- Contextual Skills. This objective focuses on the learner's acquisition of a knowledge base's organization and accessibility (i.e., contextual knowledge). The organization of a knowledge base refers to the schematic structure of the information whereas the accessibility refers to the executive control strategies that provide the means necessary to employ the knowledge base in the service of recall, problem solving, and creativity. Contextual knowledge includes the criteria, values, and appropriateness of a given domain's schematic structure. For example, simply knowing how to classify examples or

knowing how to use a rule (or principle) does not imply that the learner knows when and why to employ specific concepts or rules.

The above defined learning objectives are an extension of Gagne's conditions of learning by separating out contextual skills from his broad category of cognitive strategies. In Tennyson and Rasch's full model, cognitive strategies refer to the employment of knowledge in the service of recall and problem solving. They further extend Gagne's cognitive strategies category of objectives to include creative processes. Also, unlike Gagne's system that continues the practice of separating the cognitive and attitude domains, they include attitudes into the contextual skills category. This inclusion of the attitude within the cognitive domain is consistent with contemporary cognitive psychology (e.g., Glaser, 1990).

Information Analysis

An important component of ISD is the analysis of the information-to-be-learned. Two standard types of analyses include: (a) a content analysis, that focuses on defining the critical features of the information and the relationship of those features according to superordinate and subordinate organizations; and (b) a task analysis, that focuses on a hierarchial organization of the information based on prerequisites. Both of these analyses identify the external structure of the information but do so independent of how it might actually be stored in human memory. However, research in

cognitive psychology on human memory suggests that the internal organization of information in a knowledge base is based more on employment needs than by attribute or hierarchial associations. That is, the utility of the knowledge base is attributed to its organization not the amount of information. The implication of knowledge base organization is the need for a further analysis of the information to better understand the possible internal organization of the information. Better organization in memory may also imply better accessibility within the knowledge base for such higher order cognitive activities as problem solving and creativity.

To understand the nature of knowledge base organization, cognitive psychologist analysis problem complexity and the way individuals try to solve given problems. By analyzing problems, it is possible to identify the concepts employed; and, by analyzing the solutions, it is possible to identify the associations of those concepts within given problem situations. The implication for ISD is that the sequence of information for instruction should be based in part on internal associations as well as external structures. The assumption is that because external structures are independent of employment needs, an analysis of possible internal associations would improve a learner's initial organization of the new information.

In addition to the analyzing of problems and solutions, is the issue of problem context. For example, expert systems reside within the constraints of a specific context: That is,

they can solve problems only associated with that given context. Likewise, research in cognitive psychology shows that individuals can solve complex-problems only if they poses the necessary contextual knowledge (i.e., knowledge of when and why). For example, the objective in learning to play chess is the learning of problem solving strategies within the context of both the given game and the current move: not just how the various chess pieces move (i.e., procedural knowledge). Thus, the key to both effective acquisition and employment of knowledge is the organization of the information according to contextual applications.

The change for content/task analysis suggested by cognitive science is the method of information analysis. In addition to the conventional content and task analyses, a context analysis is proposed if the goal of the instruction includes solving complex-problems. Basic steps for a context analysis are as follows:

-Define the context for the employment of the information-to-be-learned. A context is a meaningful application of the information. The student should understand the situation presented in the context.

-Define the complex-problems associated with the context. This step follows a knowledge engineering approach where problems associated with the context are identified.

-Analysis solutions to identify concepts, rules, principles and facts employed.

-Organize the identified concepts into an associative network. Concepts which are shared by a number of problems should be taught first, to strengthen the associative network.

-Sequence the clusters into instructional components, by grouping problems according to shared concepts.

Analyzing problems within a context and then identifying the concepts and their employment organization provides a means for sequencing the instruction to improve higher order cognition. In other words, the sequence of the instruction is based on the objective of improving employment of knowledge in addition to improvements in acquisition.

This area of context analysis is a direct update of previous ISD authoring activities for content/task analyses, thus it puts AIDA1 immediately into a position of reflecting a cognitive approach to ISD. Given the steps defined above, it would be possible to implement this activity into the prototype without a time consuming effort. (I am aware that this context analysis is in direct contrast to Merrill's proposed ID Expert system, which requires an extensive analysis of "all" factors associated with a content's information base. My view is that only a basic structure needs to be identified and that the learner will fill-in the knowledge base with additional experiences during employment).

Example of A Context Analysis

The following example is presented to illustrate the above defined procedures for a context analysis. The example is

taken from a program in business management. The project used a context analysis to design an instructional program to improve problem-solving in an operations management environment. The example will follow the steps defined above.

Step 1: Define the context.

Using a simulation for the management of a kitchen cabinet factory, the student makes operational decisions which affect the profit or loss of the company. Based on the context analysis, three instructional modules were developed to prepare the student to solve problems commonly encountered during the simulation.

Step 2: Define the complex problems.

Using a knowledge engineering approach, problems were identified as representative of the situations encountered in the management of the factory. The problems were then rank-ordered by complexity; complexity being determined by the number of relevant principles required to solve the problem.

Step 3: Analyze the solutions.

Initially there were a large number of problems identified. After assigning principles to each problem, many of the problems were dropped from the list because the particular grouping of principles involved was already related to another problem. The remaining smaller group (ten problems) was then determined to represent the knowledge necessary to manage the factory. Relevant principles were identified for each problem. More complex problems required more principles to be employed in

the solution of the problem and most of the principles were used in the solution of several problems.

Step 4: Organize clusters into associative network.

Figure 2 illustrates the grouping of problems by their associated principles. The instructional design focuses on the related principles for specific problems and on shared principles which provide context for problems. That is, for each specific problem the focus is on the related principles used to solve the problem and their relationships. Principles which are used for several problems (shared principles) provide more context for the problems. As shown in Figure 2, the principles required to solve a problem are grouped according to their association.

	Problem	Principles
Module 1	Wages too low or too high	A - Wage effects on productivity
Module 1	Innappropriate rate of advertising	B - Role of advertising C - Demand D - Market Saturation
Module 1	Machine maintenance	E - Replacement of older machines F - Optimum level of repair
Module 2	Inappropriate number of machines	G - Production goals H - Number of workers I - Machine types
Module 2	Raw materials not available or price too high	G - Production goals J - Inventory K - Raw material orders L - Price differences of raw material orders
Module 2	Inappropriate number of workers	G - Production goals M - Number of machines

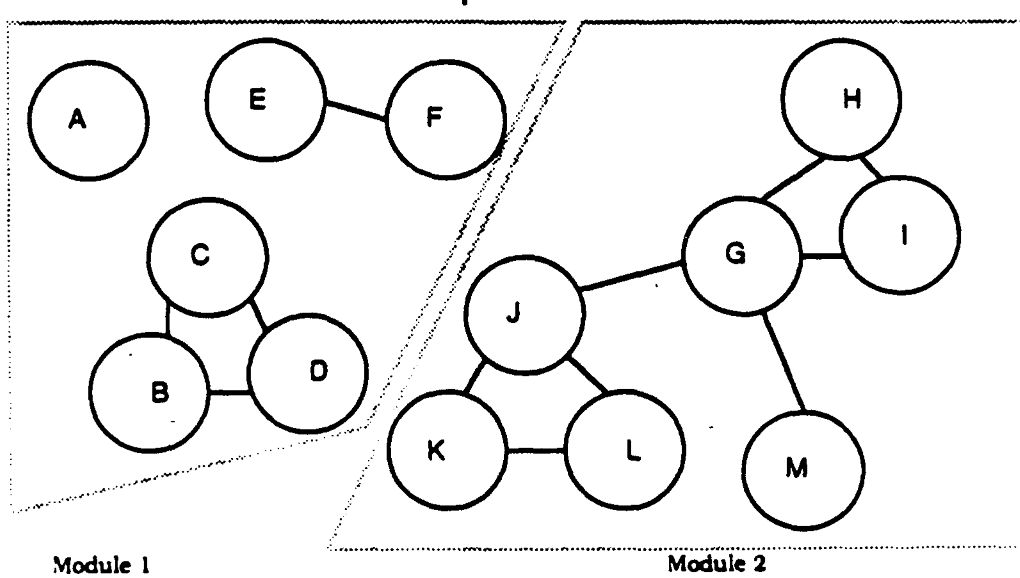

Figure 2. Problems grouped by associated principles

Step 5: Sequence network into instructional components.

Figure 3 shows the ten problems divided into three instructional modules. As you can see from the figure, the problems in the first two modules are less complex, yet most of the principles are being introduced for the first time in these problems. The problems in module three are more complex, but all the principles except one have been used in previous problems.

Module	Problem	Principles
Module 1	Wages too low or too high	A - Wage effects on productivity
Module 1	Inappropriate rate of advertising	B - Role of advertising C - Demand D - Market Saturation
Module 1	Machine maintenance	E - Replacement of older machines F - Optimum level of repair
Module 2	Inappropriate number of machines	G - Production goals H - Number of workers I - Machine types
Module 2	Raw materials not available or price too high	G - Production goals J - Inventory K - Raw material orders L - Price differences of raw material orders
Module 2	Inappropriate number of workers	G - Production goals M - Number of machines
Module 3	Production capacity not consistent with demand	F - Optimum level of repair G - Production goals H - Number of workers I - Machine types M - Number of machines
Module 3	Production and inventory not consistent with demand	B - Role of advertising G - Production goals H - Number of workers J - Inventory M - Number of machines N - Selling price
Module 3	Raw materials on hand not consistent with production goals	G - Production goals H - Number of workers K - Raw material orders L - Price differences of raw material orders M - Number of machines N - Selling price
Module 3	Demand is too low	B- Role of advertising D - Market saturation N - Price

Figure 3. Module organization by associated principles

Instructional Strategies

The purpose of the Tennyson and Rasch (1988; 1990) model is to illustrate the direct linkage of instructional strategies to specific memory system components. Also, instead of prescribing a given strategy of instruction for all forms of learning, they have identified several categories of strategies, each composed of variables and conditions that can be manipulated according to given learning situations. Within the context of this report, I will extend their general descriptions of the various categories. This extension will help in the next step of defining production rules for AIDA.

The three instructional strategy categories for AIDA1 are as follows:

Expository strategies. This category represents those instructional variables designed to provide an environment for learning of declarative knowledge (see Figure 1). The basic instructional variables provide a context for the to-be-learned information. That is, the concept of advance organizers is extended by presenting a meaningful context for the information as well as a mental framework of the given domains abstract structure. In addition to providing a context for the information, meaning can be further enhanced by adapting the context to individual student background knowledge (Ross, 1983).

The context establishes not only the initial organization of the domain but, also, introduces both the "why" of the theoretical nature of the information and the "when" of

the criterion nature of the domains standards, values, and appropriateness. Personalizing the context to student background knowledge improves understanding of the information by connecting, within working memory, knowledge that is easily retrieved. Thus, the new knowledge becomes directly linked or associated with existing schemas.

Following the contextual introduction of the information, additional expository instructional variables present the ideas, concepts, principles, rules, facts, etc. in forms that extend existing knowledge and that aid in establishing new knowledge. These variables include:

-Label. Although a simple variable, it is often necessary to elaborate on a label's origin so that the student is just not trying to memorize a nonsense word.

-Definition. The purpose of a definition is to link up the new information with existing knowledge in long-term memory; otherwise the definition may convey no meaning. That is, the student should know the critical attributes of the concept. To further improve understanding of the new information, definitions may, in addition to presentation of the critical attributes (i.e., prerequisite knowledge) include information linked to the student's background knowledge.

-Best Example. To help students establish clear abstractions of a domain's concepts, an initial example should represent an easy comprehension of the given

concept (or rule, principle, idea, etc.).

Additional expository examples will enhance the depth of understanding.

-Expository Examples. Additional examples should provide increasingly divergent applications of the information; perhaps also in alternative contexts.

-Worked Examples. This variable provides an expository environment in which the information is presented to the student in statement forms that elaborate application. The purpose is to help the student in becoming aware of the application of the information within the given context(s). For example, to learn a mathematical operation, the student can be presented the steps of the process in an expository problem while, concurrently, presenting explanations for each step. In this way, the student may more clearly understand the procedures of the mathematical operation without developing possible misconceptions or overgeneralizations.

Practice strategies. This category of instructional prescriptions contains a rich variety of variables and conditions which can be designed into numerous strategies to improve learning of procedural knowledge. This category is labelled practice, because the objective is to learn how to use knowledge correctly; therefore, it requires constant interaction between student learning (e.g., problem solving) and instructional system

monitoring. Practice strategies should attempt to create an environment in which (a) the student learns to apply knowledge to previously unencountered situations while (b) the instructional system carefully monitors the student's performance so as to both prevent and correct possible misconceptions of procedural knowledge.

The basic instructional variable in this strategy is the presentation of problems that have not been previously encountered (see Tennyson & Cocchiarella, 1986, for a complete review of variables in this category). Other variables include means for evaluation of learner responses (e.g., pattern recognition), advisement (or coaching), elaboration of basic information (e.g., text density, Morrison et al., 1988), format of information, number of problems, use of expository information, error analysis, and lastly, refreshment and remediation of prerequisite information.

Problem-oriented strategies. A proposed instructional strategy for this category uses problem-oriented simulation techniques (Breuer & Kummer, 1990). The purpose of simulations is to improve the organization and accessibility of information within a knowledge base by presenting problems that require the student to search through their memory to locate and retrieve the appropriate knowledge to propose a solution. Within this context, the simulation is a problem rather than an expository demonstration of some situation or phenomenon.

Problem-oriented simulations present domain specific problem situations to improve the organization and accessibility of information within the knowledge base. Basically, the strategy focuses on the students trying to use their declarative and procedural knowledge in solving domain-specific problems. Problem-oriented simulations present problem situations that require the student to (a) analyze the problem, (b) work out a conceptualization of the problem, (c) define specific goals for coping with the problem, and (d) propose a solution or decision. Unlike problems in the practice strategies that focus on acquiring procedural knowledge, problem-oriented simulations present situations that require employment of the domain's procedural knowledge. Thus, the student is in a problem solving situation that requires establishing connections and associations among the facts, concepts, rules, and principles of specific domains of information.

Example of an Integrated Instructional Strategy

In extending the example from the business management project, the instruction is presented in three modules, organized by grouping problems sharing common principles into an associative network. The number of modules was determined by the number of problem sets which could be identified by their common principles. The instruction is presented by (a) establishing the sub-context for the content in each module, (b) presenting the concepts in an expository manner with practice problems employing the principles in a limited context, and (c) providing a problem-

oriented simulation limited to the problems and principles presented in that module.

The instructional program was developed using the PCD3 authoring system, which uses icons to illustrate the overall instructional design. Figure 4 shows the structure of module 1, in which the material is presented first in an expository manner, with worked examples, followed by practice problems.

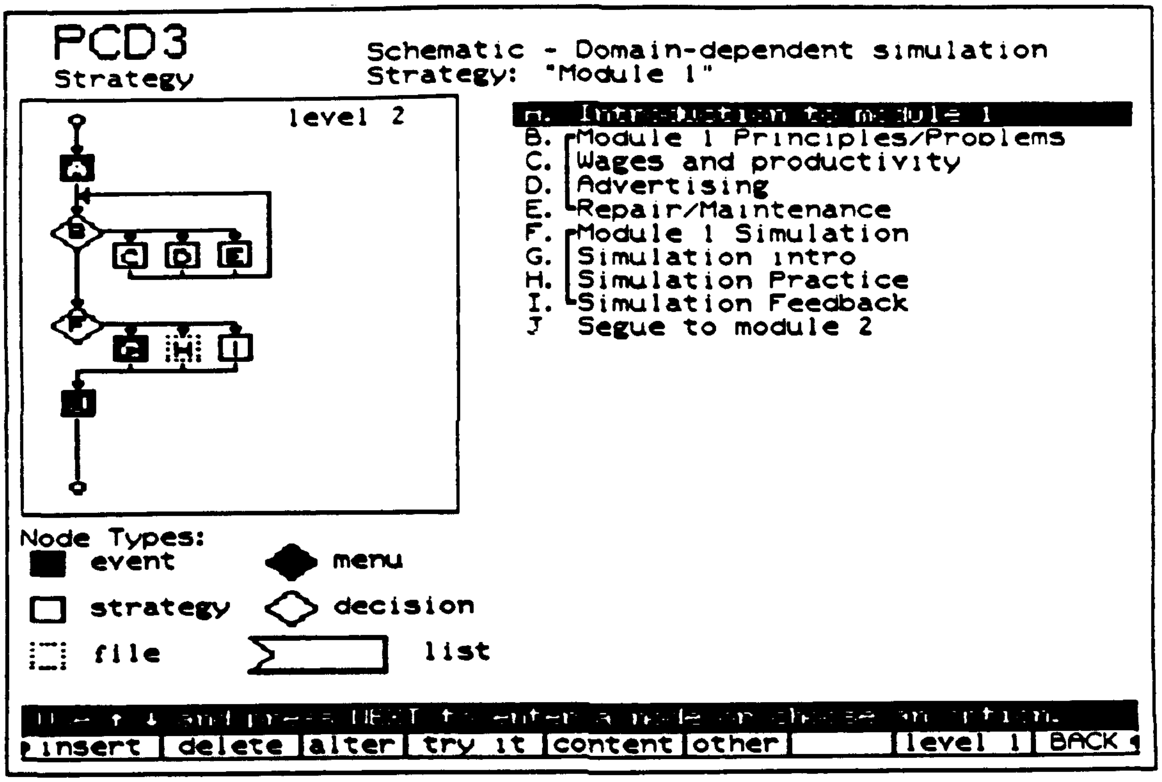

Figure 4. PCD3 strategy for module 1

Figure 5 shows an example of an expository screen using the instructional variable, worked example. Following a series of worked examples, the program continues with next instructional strategy area, practice problems.

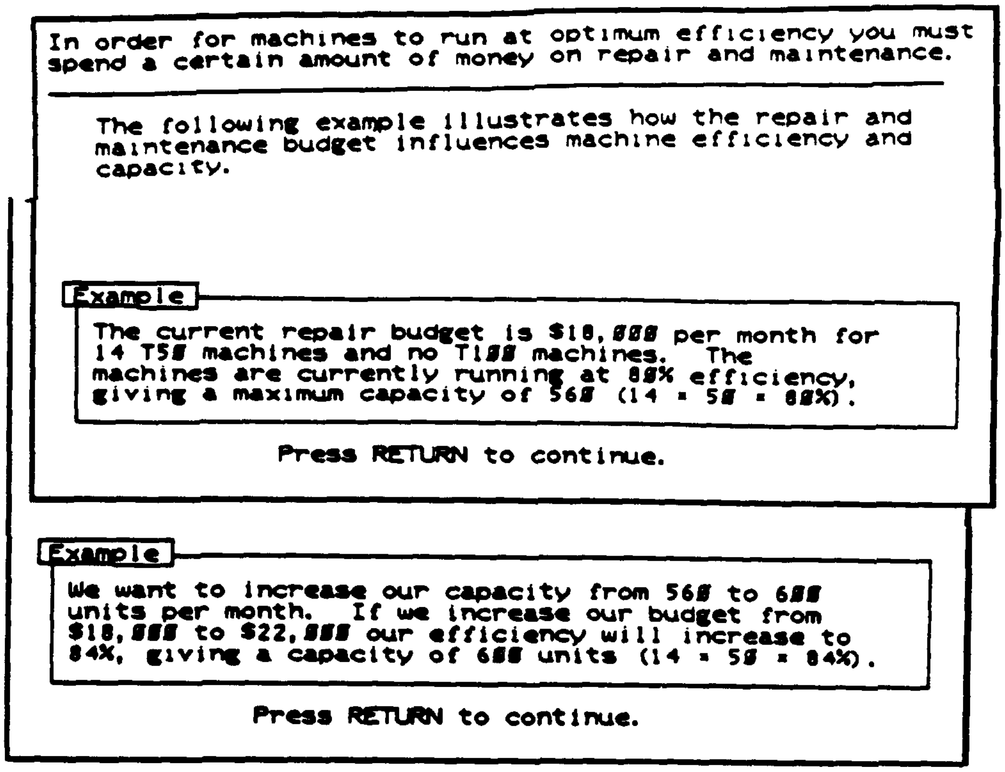

Figure 5. Expository screen with worked example

Figure 6 shows a practice problem in which the student is able to employ a set of principles in an isolated context. Principles are like rules-of-thumb, and consequently there is generally not one correct application of a principle, but effective applications fall into ranges. The practice problems allow the student to identify these ranges, based on the feedback given, and select values within them to make good decisions. As principles are combined into more complex problem solutions correct response ranges vary according to the inter-relationships of the principles involved.

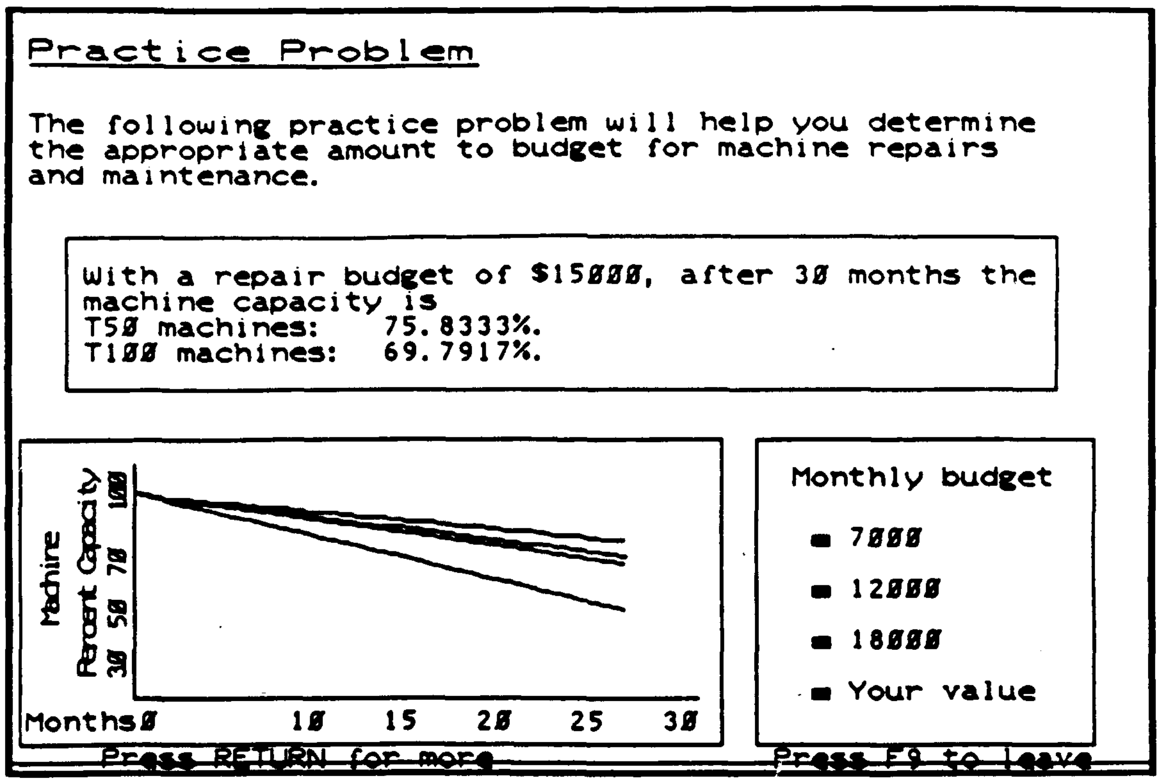

Figure 6. Practice problem

For example, the effects of advertising in an isolated context are relatively easy to observe; more advertising leads to more demand. However, the selling price of the product also affects demand and the available production capacity may not be able to accommodate an increased demand. An understanding of the effects of advertising in relation to there principles is more important than a declarative knowledge of the simple effects of advertising. As more of these principles are introduced into the context, problem solutions require an understanding of the principles and their effects, rather than learned values only.

At the end of each module the student is branched to the management simulation, but only allowed to make decisions which require principles covered in the module. All other values are held at constant levels. This allows the student to see the inter-relationships of the selected principles in the context of the simulation, but isolated from decisions related to other principles.

Figure 7 shows an example of the choices given to the student in the simulation for module 1. These choices correspond to the problems and related principles covered in this module. The student is branched through three planning periods (months) and then returned to the instruction to begin module two. At the end of each planning period, the student is given detailed information (see Figure 7) about the performance of the company during that period.

```
                                   Team: STUDENT

              DECISION   MENU
            :Wages
            :Advertising budget
            :Repair budget
            :No further decisions

 Team: STUDENT     Financial status of Company for month   1
------------------------------------------------------------------------------
                                 Gain  and  Loss

Income         : 628 units sold, priced @ $ 260 per unit          $      163280
                 Interest                                                  2096
                                                                  -------------
                                                 Income      : $        165376

Expenditures: 500 units raw material                              $       50000
              Wages for 14 workers  (wage per worker $2225/mo)            31150
              Fringe benefits (per worker $1200/mo)                       16800
              Repair budget                                               18000
              Advertising budget                                          18000
              Other expenditures                                          20000
              Storage cost for 41 units finished product                    492
                                                                  -------------
                                              Expenditures : $          154442

                                                     Gain : $            10934
                                                                      =========

------------------------------------------------------------------------------
              Balance sheet                   Start new plan cycle

  Team: STUDENT     Financial status of Company for month   1
------------------------------------------------------------------------------
Assets                         Monthly Statement                    Liabilities
------------------------------------------------------------------------------
Fixed Assets:                                / Capital resources: $      623252
  14 Type 50 mach @ $44518     $      623252/
                                             / Open reserves               5125
Liquid assets:                               / Reserves                   75000
  Inventory                                  /
                                             /
    Finished product 41 units          5125/
  Cash on hand:                       85934/ Gain                         10934
                                  ---------/                          ---------
Total assets:                  $     714311/ Total liabilities: $        714311
------------------------------------------------------------------------------
14 workers for 14 Type 50 machines           /
                                             /
average capacity Type  50 mach:  92.75 %     /
                                             /
                                             /
Production goal              :  625 units  / Raw material price:     $    136
Production                   :  625 units  /
Demand in constant  market   :  628 units  /
------------------------------------------------------------------------------
                Gain and Loss       Start new plan cycle
```

Figure 7. Module Simulation

At the conclusion of the three instructional modules (including three module simulations) the student is branched to the simulation again. At this point the student is required to make all the decisions related to the management of the factory for twelve planning periods. Figure 8 shows the decisions the student makes in the complete simulation. Because of the increased complexity of the complete simulation, twelve cycles are necessary to allow students to encounter problems and follow through on solution strategies. Again, the student is given detailed information at the conclusion of each planning period (see Figure 8).

```
Z0000000000000000000000000000000000000000000000000000000000000000000000000000000?
3                                  Team: STUDENT                                  3
3                                                                                 3
3                                                                                 3
3                                                                                 3
3                                                                                 3
3           :MMMMMMMMMMMMMM DECISION  MENU OMMMMMMMMMMMMMM:                       3
3           :                                                :                    3
3           :Raw material orders and spot market buys        :                    3
3           :Production goal changes                         :                    3
3           :Selling price changes                           :                    3
3           :Machine purchases and sales                     :                    3
3           :Allocation of workers                           :                    3
3           :Repair budget                                   :                    3
3           :Wages                                           :                    3
3           :Advertising budget                              :                    3
3           :No further decisions                            :                    3
3           HMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM<                    3
3                                                                                 3

 Team: STUDENT    Financial status of Company for month  1
*****************************************************************************
Assets                          Monthly Statement                  Liabilities
------------- -----------------------------------------------------------------
Fixed Assets:                                / Capital resources: $      623252
  14 Type 50 mach @ $44518    $      623252/
                                             / Open reserves             43750
Liquid assets:                               / Reserves                  75000
  Inventory                                  /
    Raw material 875 units           43750/

  Cash on hand:                     139426/ Gain                        64426
                                 *********/                         *********
Total assets:                 $     806428/ Total liabilities: $            0
((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
12 workers for 14 Type 50 machines          /Arriving in   Qty          Price
                                            /month   2       0 units  $   100
average capacity Type  50 mach:  92.75 %   /
                                            /
                                            /
Production goal               :  600 units / Raw material price:      $   137
Production                    :  594 units / Spot mkt supply:       122 units
Demand in constant  market    :  687 units / Spot mkt price :         $   151
*****************************************************************************
                  Gain and Loss        Make decisions

 Team: STUDENT    Financial status of Company for month  1
*****************************************************************************
                             Gain  and  Loss

Income        : 638 units sold, priced @ $ 250 per unit          $      159500
                interest                                                  3401
                                                                 -------------
                                                    Income    : $      162901

Expenditures: Wages for 12 workers  (wage per worker $2225/mo)           26700
              Fringe benefits (per worker $1200/mo)                      14400
              Repair budget                                              18000
              Advertising budget                                         18500
              Other expenditures                                         20000
              Storage cost for 875 units raw material                      875
                                                                 -------------
                                               Expenditures : $          98475

                                                        Gain : $         64426
                                                                     *********


*****************************************************************************
                Balance sheet                        Make decisions
```

Figure 8. Full Simulation

Conclusion

This Cycle 1 report presents a modified set of ISD authoring activities for an initial AIDA prototype. It follows my recommendation from my Phase 1, Cycle 2 report that the development of AIDA follow an incremental approach. The authoring activities selected for this prototype were taken from my Phase 1, Cycle 1 updated ISD model. They were selected to provide a basic set of authoring activities associated with developing instruction for the acquisition of knowledge. Also, because this prototype is not intended for actual use, I am recommending that it be designed for experienced courseware authors. As more research is done, subsequent AIDAs would exhibit more intelligence to account for less-experienced authors.

Three sets of authoring activities were identified as representing a minimal program of ISD. These three activities are necessary for the development of a piece of computer-based instruction. They include identifying the learning objectives, analysis of the information to be learned, and selection of appropriate instructional strategies. Also, given the nature of AIDA to represent contemporary research in learning and instruction, the activities are cognitive in foundation rather than behavioral. This is an important aspect of the proposal because the activities do differ from current ISD practice.

The learning objectives are an extension of Gagne's basic conditions of learning in two important ways. First, the

category of contextual skills is in response to cognitive concepts associated with situated cognition and constructionism. Thus, it separates out that higher-order (i.e., contextual) knowledge from Gagne's broader category labelled cognitive strategies. Second, contextual skill objectives reflect in part the attitudinal domain. Therefore, instead of separating the cognitive and attitude domains, contextual skill objectives combine them.

Information analysis proposes an additional analysis of the information based upon complex-problems associated with a given context. Whereas, conventional content/task analyses identify the attributes of the information, the context analysis identifies the schematic organization of the information. The schematic organization improves the service of the knowledge base for higher order employment situations (i.e., problem solving and creativity).

The third set of authoring activities directly links cognitive learning theory with specific instructional strategies. Rather than acquisition of knowledge in isolation, it is proposed that learners acquire knowledge within meaningful situations. Unfortunately, research in instructional design theory has focused on strategies associated with declarative and procedural knowledge with minimal empirical work for strategies associated with contextual knowledge. The instructional strategies provide an opportunity to develop instruction at each of the three main categories or any number integrated strategies.

The next section by Reigeluth was originally 2 separate papers. In the first paper (Part 1), Reigeluth gives general simplifying conditions and functions important for development of a potential lesson for an XAIDA. In Part 2 he applies the conditions and functions to the development of an actual lesson, Engine Startup Procedures for the T-38.

III. EPITOMIZING AIDA (Reigeluth)

General Simplifying Conditions and Functions for XAIDA

Simplifying Conditions. The functions that the XAIDA should include are indicated by the simplifying conditions that are selected for it. I propose the following simplifying conditions:

- The task is a simple procedural one with an orientation to equipment.
- The task takes less than 10 hours to teach.
- The task does not require the teaching of generic skills (cognitive strategies), or attitudes/values.
- The medium is computer-based instruction, and more specifically programmed tutorial and simulations.
- The environment is a computer learning center with a proctor.
- The general instructional strategy is restricted to expository, or didactic, instruction.

Functions. With these simplifying conditions in place, the following are the functions that the system must possess to perform its designated job. Note that the functions are defined by their output, rather than by their input or process. Often several different inputs and processes are needed to perform one of these functions.

1. Confirmation of Sequencing Strategy.

    Output: Confirmation that the task is procedural and short; Selection of the template for "short procedure" task analysis.

1. Confirmation of Sequencing Strategy.

Output: Confirmation that the task is procedural and short; Selection of the template for "short procedure" task analysis.

Process: XAIDA prompts the instructional designer to interview a task expert to find out how much variation there is from one performance of the task to another, and to find out whether the task can be easily proceduralized.

Input: A task expert.

2. Task Analysis

Output: A list at entry level of description, of the steps that need to be taught, in the order in which they need to be performed, for a target learner to become an expert at this task.

Process: XAIDA presents a template for the designer to fill in while interviewing a task expert. The template prompts the designer in the performance of a procedural and hierarchical task analysis, and in the performance of relevant aspects of learner analysis. It also prompts the designer on how to confirm the results of the analysis with other task experts and designer observation of the task.

Input: Job situation: Several task experts; Several marginal target learners (lowest entering ability) or

an instructor very familiar with their entry-level abilities.

3. Macro Design I

Output: All entry-level steps arranged in a procedural sequence.

Process: XAIDA executes an algorithm which prepares an outline of the procedural sequence for teaching the task. (Steps are sequenced in the order in which they are performed on the job.) The designer can modify the outline as he or she sees fit.

Input: Output of Function 2.

4. Macro Content Analysis

Output: List of all supporting content to be taught with each step of the procedure. This includes learning prerequisites, relevant principles and concepts, and useful information.

Process: XAIDA prompts the designer to fill out templates with a task expert. The templates are based on the Elaboration Theory's content analysis procedures, including Gagne's hierarchical analysis, and on Merrill's recent work. The designer can also modify any template for any given step.

Input: Output of Function 2; Several task experts; Several marginal target learners or an instructor very familiar with their entry-level abilities.

5. Macro Design II

<u>Output</u>: A clustering of all steps and their related content into instructional modules; An outline for a "nested sequence" for each step and its related content within each module.

<u>Process</u>: XAIDA prompts the designer to decide how long a module should be and to allocate steps and their related content to modules. Then it applies rules (asking questions of the designer or task expert when necessary) to generate on outline for a within-module sequence of content for each module. The designer can modify the content outline for any module as he or she sees fit.

<u>Input</u>: Output of Function 4; Information about the environment and learners to decide on the optimal size of a module.

6. Micro Content Analysis

<u>Output</u>: Classification of level of learning (memorization or application) for each piece of content in each module.

<u>Process</u>: XAIDA presents default classifications (based on a few simple decision rules) for each piece of content, and asks for confirmation from the task expert, based on that person's determination of post-instructional requirements. It then requests

confirmation from a second task expert, time permitting.

Input: Output of Function 4; Several task experts.

7. Selection of Appropriate Template

Output: Allocation of a template to every piece of content that has been selected for instruction.

Process: XAIDA uses a simple matching algorithm.

Input: Output of Function 6.

8. Micro Content Analysis II

Output: Assessment of level of learning difficulty for each piece of content selected for instruction: Identification of dimensions of divergence for each skill selected for instruction.

Process: For each piece of content, XAIDA prompts the designer to elicit learning difficulty levels (on say a scale of 1-5) from an instructor familiar with the target learners. It also requests confirmation from a second instructor, time permitting. (later, during the formative evaluation, it will test and revise those estimates.) For each skill, XAIDA prompts the designer to elicit dimensions of divergence and the important variations for each dimension from a task expert. It also requests confirmation from a second task expert, time permitting.

Input: Output of Functions 4, 6, and 7; Several task experts; Several instructors familiar with the target learners.

9. Micro Design Specifications

Output: Expansions of the templates from Function 7, to include slots for tactics appropriate for the designated level difficulty, and slots for examples and practice for each variation of each dimension of divergence; A template for a proctor's guide for each module.

Process: XAIDA uses production rules to expand existing templates (from Function 7) and it uses decision rules to replace existing templates (from Function 7) with more elaborate templates. It also uses matching algorithms to select appropriate templates for the proctor's guide for each template for the instruction.

Input: Outputs of Functions 7 and 8.

10. Develop Instructional System

Output: A complete instructional system, including computer-based instruction, tests, and proctor's guide.

Process: XAIDA prompts the task expert, under the watchful eye and clarifications of the designer, to fill in all templates with words and graphics, modifying any templates as they see fit. XAIDA automatically programs and compiles the CBI and

automatically prints out the proctor's guide after they have been reviewed and confirmed by other task experts.

Input: The output Functions 5 and 9; Several task experts.

11. Formative Evaluation and Revision

Output: A revised instructional system that is proven effective.

Process: XAIDA collects data from learners from the target population as they proceed through the instruction. It identifies weak points in the instruction, and proposes solutions for approval or modification by the designer and task experts. All approved solutions are automatically made by XAIDA, the program is recompiled, and the proctor's manuals reprinted.

Input: The output of Function 10; Several learners from the target population; Several task experts.

XAIDA Functions for the Instruction of the T-38 Engine Startup Procedure

1. Confirm Sequencing Strategy

   Input: An expert in maintenance of the T-38 engine starting system.

   Process: XAIDA prompts the instructional designer to interview a task expert to find out whether the task can be easily proceduralized.

   Output: Confirmation that the task is procedural; Selection of the template for "procedure" task analysis.

2. Identify a "Just Simple Enough" Class of Cases

   Input: Job situation; Several task experts; Several marginal target learners (lowest entering ability) or an instructor very familiar with their entry-level abilities; Information about the environment and learners to decide on optimal size of a module.

   Process:

   - XAIDA prompts the designer to have the task expert think about what makes some "engine starting system maintenance" cases easier than others, and then to think of the simplest class of cases he or she ever performed.
   - It then prompts the designer to develop (with an experienced instructor) an estimate as to how many hours of intensive learning time it would take a target

learner to learn to perform that simplest class of cases as an expert would perform it (including time to develop all necessary mental models).

- XAIDA prompts the designer to decide how long a module of a training course should be (approx. 3-10 hours of learning time). If the amount of time required is too long or too short for a single module, XAIDA prompts the designer and task expert to further simplify or expand the simplest class of cases.
- Then XAIDA prompts the designer and task expert to list the conditions that make the simplest class simpler than the most complex class of cases.

Output: Identification of simplest class of cases and its simplifying conditions:

- Simplest class: Restart left engine when right engine is running.
- Simplifying conditions: Right engine already running (no need for power connections), Functional testing (no exterior or interior inspections needed), No engine testing needed, No troubleshooting needed, No emergency procedures needed, ...

3. Identify Progressively More Complex Classes of Cases (Can be done after 4)

Input: Simplifying conditions (output of Function 2).

Process: Rank order the simplifying conditions on the basis of how important and representative of the whole task its corresponding class of cases is.

Output: A simple-to-complex sequence of classes of cases for the task (it's hard for me to do this without a task expert to question):

- Start right engine when plane is on ground (requires power hook-ups).
- Restart engine after maintenance (requires interior and exterior inspections).
- Engine problems--diagnosis [Section IV]: operating limits, instrument tolerance,... (requires diagnosis procedures).
- Engine maintenance testing [Section V] (requires operating tests and inspections).
- Engine troubleshooting [Section VI]: No start, slow start, RPM hang-up, hot start (requires troubleshooting procedures).
- Engine problems--emergencies: fire, overtemperature, overspeed, smoke/fumes, oil system, generator, hydraulic system, compressor stall, engine flameout (requires emergency procedures).

4. Conduct Task Analysis on Each Class of Cases

Input: Output of Functions 2 and 3; Job situation; Several task experts; Several marginal target learners

(lowest entering ability) or an instructor very familiar with their entry-level abilities.

Process: For each class of cases, XAIDA presents templates for the designer to fill in while interviewing a task expert.

- There is a device template which prompts the designer to input a diagram of each device (or each variation of a device) operated upon in performance of the task for the simplest class of cases.
- Then XAIDA prompts the designer to identify (from the task expert) the alternative procedures that an expert would use to perform the simplest class of cases, and to input a label for each. (In some situations there may be only one alternative.)
- Then XAIDA sets up a procedure template for each alternative procedure and prompts the designer to conduct a procedural task analysis with the task expert) to fill in the template. This analysis identifies all steps (at entry level of description) in each alternative procedure, in the order in which they need to be performed, along with the objects (or parts of devices) that are acted upon in each step of the procedure and the tools that are required in performing each step.
- Using the results of the procedure analysis, XAIDA generates Kinds taxonomies and parts taxonomies for the

objects and tools, using appropriate templates. The task expert is asked to modify and expand them as appropriate. It also develops a graphic physical model of each device, by asking the task expert to label each of the parts (objects) on the diagrams entered earlier.

- There is a functional model template (referred to by Henry Halff as a conceptual model) which prompts the task expert to create a schematic representative of how each device works. It can also be applied to tools when appropriate.
- XAIDA prompts the designer on how to confirm the results of the analysis with other task experts and designer observation of the task.

<u>Output</u>: A procedural model for each alternative procedure, kinds and parts taxonomies for all objects and tools, and a physical model and a functional model for each device (and each tool as appropriate), all validated by several experts.

For simplest class: Restart left engine when right engine is running:

- Procedural model [2-6]: 1) Clear danger areas; 2) Signal ground crewman to apply external air; 3) Push engine start button momentarily; 4) Advance throttle to idle at 14% min. RPM; 5) Check...; 6)...; 7)...

- Taxonomies: Certain kinds of instruments and controls in cockpit (only those that will be used during the procedure).
- Physical model for each device: Cut-away drawing of parts of airplane the learner will be using in the simplest class of cases.
- Functional model for each device: Schematic drawing (preferably dynamic) of parts of airplane the learner will be using in the simplest class of cases.

5. Design the Sequence of Major Content

<u>Input</u>: Output of Function 4.

<u>Process</u>: XAIDA executes an algorithm which prepares an outline of the sequence for teaching each class of cases. For a given class of cases, one alternative procedure is picked for one kind of device, and:

1. The functional model for that device comes first.
2. The physical model for that device (including all of its parts or objects) comes next and is related to the functional model.
3. The parts taxonomy for the device comes next (as a synthesizer).
4. The procedural model comes next, with its entry-level steps sequenced in the order in which they are performed on the job, and each tool being listed joust before it is needed in the procedure.

5. The remaining taxonomies are presented as synthesizers. The same kind of sequence is outlined for each additional device and procedure for this class of cases, and for each subsequent class of cases. The designer and task expert can modify the outline as they see fit.

<u>Output</u>: An outline of the sequence for all major content to be taught for all classes of cases. For simplest class of cases: Since there is only one functional model, one physical model, one taxonomy, and one procedural model, the sequence is as outlined under process above.

6. Analyze Supporting Content (Can be done after Function 3 or 4)

<u>Process</u>: For each class of cases, XAIDA prompts the designer and task expert to fill out slots for supporting content. The supporting content includes primarily principles, attitudes, information, and prerequisite concepts and discriminations. The templates are based on the Elaboration Theory's content analysis procedures, including Gagne's hierarchical analysis, and on Merrill's recent work. The designer can also modify any template for any given step.

<u>Output</u>: List of all supporting content to be taught with each step of the procedure. This includes learning prerequisites, relevant principles and concepts,

related attitudes and values, and useful information. [For the T-38 engine restarting task, I have run out of time and cannot quickly find enough task knowledge to perform this or any of the remaining functions.]

7. Design a Content Sequence for Each Module

<u>Input</u>: Output of Function 5.

<u>Process</u>: - Bases on the earlier decision about how long a module should be, XAIDA prompts the designer to allocate major content and its related supporting content to modules, using estimates from an experienced instructor as to how long it will take to teach the content.

-Then XAIDA applies rules (asking questions of the designer or task expert when necessary) to generate an outline for a within-module sequence of content for each module. The designer can modify the content allocation and module sequences as he or she sees fit.

<u>Output</u>: A clustering of all major and supporting content into instructional modules; An outline for the sequence of all content within each module, including simulations that provide integrated demonstrations or practice for the whole task or part-tasks.

8. Classify Micro Content

<u>Input</u>: Output of Function 5; Several task experts.

<u>Process</u>: XAIDA presents default classifications for type of learning (based on a few simple decision rules) for

each piece of content, and asks for confirmation from the task expert (with the help of the designer), based on that person's determination of post-instructional requirements. It then requests confirmation from a second task expert, time permitting.

Output: Classification of type of learning for each piece of content in each module.

9. Decide on a Strategy for Each Cluster of Content.

Input: Output of Functions 6 and 7.

Process: For each module, decide what will be taught by programmed tutorial, by drill and practice, and by simulation. Some content may be taught by several strategies (e.g., the procedural model may be taught via generality-demonstration-practice-feedback in a tutorial [low fidelity of representation], followed by additional demonstration-practice-feedback in a simulation). Revise the sequence of content for the module, as appropriate.

Output: Allocation of content to strategies, and revised sequence of content.

10. Select Appropriate Template for Tactics for Each Piece or Cluster of Content for Each Strategy.

Input: Output of Function 8.

Process: XAIDA uses a simple matching algorithm based on type of learning and type of strategy to select a "lean" template for each piece or cluster of content.

Output: Allocation of a lean template to every piece or cluster of content that has been selected for instruction.

11. Analyze Micro Content

Input: Output of Functions 5, 7, and 9; Several instructors familiar with the target learners.

Process: For each piece or cluster of content, XAIDA prompts the designer to elicit learning difficulty levels (on, say, a scale of 1-5) from an instructor familiar with the target learners. It also requests confirmation from a second instructor, time permitting. (Later, during the formative evaluation, it will test and revise those estimates.)

- For each skill, XAIDA prompts the designer to elicit dimensions of divergence and the important variations for each dimension from a task expert.
- For each simulation, XAIDA prompts the designer to elicit a scenario and a causal model (qualitative or quantitative) to govern the computer's actions in the simulation.
- XAIDA also requests confirmation from a second task expert, time permitting.

Output: Assessment of Level of learning difficulty for each piece or cluster of content selected for instruction; Identification of dimensions of divergence for each skill selected for instruction.

12. Modify Micro Templates

Input: Outputs of functions 9 and 10.

Process: XAIDA uses production rules to expand existing lean templates (from Function 9), including slots for examples and practice for each variation of each dimension of divergence; and it uses decision rules to replace existing templates (from Function 9) with more elaborate templates. It also uses matching algorithms to select appropriate templates for the proctor's guide for each template in the computer-based instruction.

Output: Expansions of the templates from Function 9, to include slots for tactics appropriate for higher levels of difficulty, and slots for examples and practice for each variation of each dimension of divergence; A template for a proctor's guide for each module.

13. Develop Instructional System

Input: The output of Functions 6 and 11; Several task experts.

Process: XAIDA prompts the task expert, under the watchful eye and clarification of the designer, to fill in all templates with words and graphics, modifying any templates as they see fit. XAIDA automatically programs and compiles the CBI and automatically prints out the proctor's guide after each has been reviewed and confirmed by other task experts.

Output: A complete instructional system, including computer-based instruction, tests, and proctor's guide.

14. Formatively Evaluate and Revise

Input: The output of Function 13; Several learners from the target population; Several task experts.

Process: XAIDA collects data from learners from the target population as they proceed through the instruction. It identifies weak points in the instruction, and proposes solutions for approval or modification by the designer and task experts. All approved solutions are automatically made by XAIDA, the program is recompiled, and the proctor's manual is reprinted.

Output: A revised instructional system that is proven effective.

V. CONCLUSIONS (Gettman)

This volume is dedicated to outlining the development of a prototype AIDA or XAIDA. Tennyson asserts that an incremental approach to AIDA development is optimal and in this volume he describes the basic set of authoring activities which should be included. Tennyson proposes that XAIDA development follow his "Fourth Generation" ISD model in which three basic sets of authoring activities; identification of learning objectives, analysis of the information to be learned, and instructional strategies, guide instructional development.

Reigeluth agrees that AIDA development should follow an incremental approach and in his paper, he first describes necessary "simplifying conditions" for an XAIDA. According to Reigeluth, these "conditions" are specific to this version of AIDA and may or may not be necessary for the development of succeeding versions of the design advisor. Next, Reigeluth presents the "functions" that the XAIDA must possess in order to design the designated instruction. He outlines the necessary functional steps in three phases; analysis, design, and development and evaluation.

Tennyson's cogent description of linking memory types with learning objectives will set the stage for defining functional characteristics of an instructional design advisor. Providing proper learning objectives and the context in which they will be learned will be requisite information not only for XAIDA, but for each progressive module as more sophisticated

design advisors are developed. Tennyson's instructional "rules" based on sound ISD concepts can be viewed as an elaboration of Reigeluth's "simplifying conditions."

Tennyson and Reigeluth have presented complementary views for development of an XAIDA. Tennyson's paper describes the basic paradigm to follow for development of XAIDA. Reigeluth then applies this paradigm to a practical lesson.

REFERENCES

Breuer, K., & Kummer, R. (1990). Cognitive effects from process learning with computer-based simulations. Computers in Human Behavior, 6, 69-81.

Gagne, R.M., & Merrill, M.D. (1990). Integrative goals for instructional design. Educational Technology: Research and Development.

Glaser, R. (1990). The reemergence of learning theory within instructional research. American Psychologist, 45, 29-39.

Hewett, T. T. (1989). Towards a rapid prototyping environment for interface design: Desirable features suggested by the electronic spreadsheet. In A. Sutcliffe & L. Macaulay (Eds.), People and computers V (pp. 305-314). Cambridge, UK: Cambridge University Press.

Morrison, G. R., Ross, S. M., O'Dell, J. K., Schultz, C. W., & Higginbotham-Wheat, N. (1989). Implications for the design of computer-based instruction screens. Computers in Human Behavior, 5 , 167-174.

Ross, S. M. (1983). Increasing the meaningfulness of quantitative material by adapting context to student background. Journal of Educational Psychology, 75, 519-529.

Tennyson, R.D. (1989, October). Task 2.2. Cognitive science and instructional technology updates of ISD. (Mei & Associates, F33615-88-C- 0003). Minneapolis, MN: Tennyson & Associates, Inc.

Tennyson, R. D. (1990a). A proposed cognitive paradigm of learning for educational technology. Educational Technology, 26(6), 12-17.

Tennyson, R.D. (1990b, February). Task 2.7. Framework specifications document for an instructional systems development expert system. (Mei & Associations, F33615-88-C-0003). Minneapolis, MN: Tennyson & Associates, Inc.

Tennyson, R. D., & Cocchiarella, M. J. (1986). An empirically based instructional design theory for concept teaching. Review of Educational Research, 36, 40-71.

Tennyson, R. D., & Rasch, M. (1988). Linking cognitive learning theory to instructional prescriptions. Instructional Science, 17, 369-385.

Tennyson, R.D., & Rasch, M. (1990). Instructional design for the improvement of learning and cognition. In H. Feger (Ed.), Wissenschaft und Verantwortung (pp 1-15). Goettingen, Fed. Rep. of Germany: Hogrefe-Verlag.